WHAT IS CLAIMED IS:

1. A method for an inverse-modified discrete cosine transform and overlap-add for MPEG Layer3 audio signal decoding, comprising the steps of:

applying an operation of the inverse-modified discrete cosine transform and overlap-add according to equation (1) to 32 sub-band samples of a compressed audio signal, wherein the equation (1) includes

an inverse-modified discrete cosine transform:

$$x(i)=\sum_{k=0}^{\frac{n}{2}-1}X(k)*\cos(i,k) \qquad 0\le i\le\frac{n}{4}-1 \text{ and } \frac{n}{2}\le i\le\frac{3n}{4}-1$$

and an overlap-add:

$$Z(i)=x(i)*win(i,p)$$

$$Z(\frac{n}{2}-1-i)=-x(i)*win(\frac{n}{2}-1-i,p) \qquad 0\le i\le\frac{n}{4}-1$$

$$Z(i)=x(i)*win(i,p)$$

$$Z(n-1-i)=x(i)*win(n-1-i,p) \qquad \frac{n}{2}\le i\le\frac{3n}{4}-1 \text{ ,}$$

where X(k) is the sub-band sample, Z(i) is the sub-band sample after process, when a window type is 0, 1, 3, n equals 36, and when the window type is 2, n equals 12;

providing a dynamic window inverse-modified discrete cosine transform (DWIMDCT) module, wherein a multiplier-adder of the dynamic window inverse-modified discrete cosine transform module processes an operation of the inverse-modified discrete cosine transform, and an operation result of the inverse-modified discrete cosine transform is stored in a register stack of the dynamic window inverse-modified discrete cosine transform module; and

using the multiplier-adder to operate the overlap-add operation, and an operation

result of the overlap-add is stored in a dynamic window inverse-modified discrete cosine transform buffer memory.

2. The method of claim 1, further comprising the steps of:

applying an efficient memory layout and a data arrangement method to the dynamic window inverse-modified discrete cosine transform buffer memory to store a plurality of data generated by the dynamic window inverse-modified discrete cosine transform module, to provide a reading operation of a synthesis filter bank module; and

alternately writing to and reading from the dynamic window inverse-modified discrete cosine transform buffer memory.

3. The method of claim 2, wherein the dynamic window inverse-modified discrete cosine transform module and the synthesis filter bank module can be implemented in a manner of a pipeline process.

4. The method of claim 2, wherein the dynamic window inverse-modified discrete cosine transform buffer memory comprises 3 memory banks, each of the memory banks is further divided into 32 sub-band blocks, and each of the sub-band blocks is able to store 18 sample data.

5. The method of claim 4, wherein the writing of the dynamic window inverse-modified discrete cosine transform of the sample data contained in each of the memory banks of the dynamic window inverse-modified discrete cosine transform buffer memory and the reading of the synthesis filter bank follows a specific sequence.

6. The method of claim 1, wherein the register stack comprises 18 registers.

7. The method of claim 1, wherein the method can be used in a hardware structure design of a post-process portion in an audio decoding process of a Layer3 compression method in an MPEG compression standard (MP3).

8. A hardware structure of an inverse-modified discrete cosine transform and an overlap-add for MPEG Layer3 audio signal decoding, comprising:

a dynamic window inverse-modified discrete cosine transform module, comprising:

a multiplier-adder, used to calculate the inverse-modified discrete cosine transform and the overlap-add; and

a register stack, coupled to the multiplier-adder, used to store an operation result of the inverse-modified discrete cosine transform; and

a dynamic window inverse-modified discrete cosine transform buffer memory, coupled to the dynamic window inverse-modified discrete cosine transform module, used to store an operation result of the overlap-add.

9. The hardware structure of claim 8, wherein the inverse-modified discrete cosine transform and the overlap-add are operated by equation (1) below:

the inverse-modified discrete cosine transform:

$$x(i)=\sum_{k=0}^{\frac{n}{2}-1}X(k)*\cos(i,k) \qquad 0\le i\le\frac{n}{4}-1 \text{ and } \frac{n}{2}\le i\le\frac{3n}{4}-1,$$

the overlap-add:

$$Z(i)=x(i)*win(i,p)$$

$$Z(\frac{n}{2}-1-i)=-x(i)*win(\frac{n}{2}-1-i,p) \qquad 0\le i\le\frac{n}{4}-1$$

$$Z(i)=x(i)*win(i,p)$$

$$Z(n-1-i)=x(i)*win(n-1-i,p) \qquad \frac{n}{2}\le i\le\frac{3n}{4}-1,$$

where X(k) is a sub-band sample, Z(i) is the sub-band sample after process, when a window type is 0, 1, 3, n equals 36, and when the window type is 2, n equals -12.

10. The hardware structure of claim 8, wherein the dynamic window inverse-modified discrete cosine transform buffer memory is applied with an efficient memory layout and a data arrangement method, to store a plurality of data generated by the dynamic window inverse-modified discrete cosine transform module for providing a reading of a synthesis filter bank module.

11. The hardware structure of claim 10, wherein the dynamic window inverse-modified discrete cosine transform module and the synthesis filter bank module can be implemented in a pipeline process manner.

12. The hardware structure of claim 10, wherein the dynamic window inverse-modified discrete cosine transform buffer memory comprises 3 memory banks, each of the memory banks is further divided into 32 sub-band blocks, and each of the sub-band blocks is able to store 18 sample data.

13. The hardware structure of claim 12, wherein the writing of the inverse-modified discrete cosine transform of the sample data contained in each of the memory banks of the dynamic window inverse-modified discrete cosine transform buffer memory and the reading of the synthesis filter bank follow a specific sequence.

14. The hardware structure of claim 8, wherein the register stack comprises 18 registers.

15. The hardware structure of claim 8, wherein a hardware structure can be used in a hardware structure design of the post-process portion in the audio decoding process of the Layer3 compression method of the MPEG compression standard (MP3).

16. The hardware structure of claim 8, wherein the hardware structure can be implemented by applying the application specific integrated circuit (ASIC).